UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/239,194	01/28/1999	JOHN S YATES JR.	114596-05-4013	9716

38492 7590 04/03/2006

WILLKIE FARR & GALLAGHER LLP
INTELLECTUAL PROPERTY LEGAL ASSISTANTS
787 SEVENTH AVE
NEW YORK, NY 10019-6099

EXAMINER	
TANG, KENNETH	
ART UNIT	**PAPER NUMBER**
2195	

DATE MAILED: 04/03/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/239,194	YATES ET AL.	
	Examiner	Art Unit	
	Kenneth Tang	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. §.133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 January 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-83* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-83* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *1/3/06*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail.Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the Amendment filed on 1/03/06.

2. Applicant's arguments have been fully considered but are not found to be persuasive.

3. Claims 1-83 are presented for examination.

Allowable Subject Matter

4. Claims 1-4 would be allowable if rewritten or amended to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action.

5. Claims 14, 25, 28-29, 32, 37, 47-49, 51, 53-55, 76-78, and 80-81 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims.

Response to Arguments

6. It is noted that the Applicant failed to respond and argue to the 35 USC 112, 2nd paragraph rejections from the last Office Action on paragraph 4(a)(i-iv) and 4(b)(i, ii, viii, x, xi, xii, and xv). Rule § 1.111 states that in order to be entitled to reconsideration or further examination, the applicant or patent owner must reply to the Office action. The reply by the applicant or patent owner must be reduced to a writing which distinctly and specifically points out the supposed errors in the examiner's action and must reply to every ground of objection and rejection in the prior Office action. It is noted that the Applicant does not present any arguments or response to these rejections and they are maintained.

7. Applicant argues that the new reference of Robinson is almost the same as the reference of Chernoff applied in the last office action.

In response, the Examiner doesn't understand why this is an issue to the Applicant. The Examiner is entitled to reopen prosecution with the same reference, if it was decided to. It is irrelevant whether Chernoff and Robinson are similar. The Pre-Appeal Conference's decision to reopen prosecution was so that new grounds of rejections based on 35 USC 112, 2nd paragraph (indefiniteness and lack of antecedent basis) could be introduced and applied, not because the reference of Chernoff should not have been used as prior art.

8. Applicant argues regarding the rejection of claim 5, that Robinson discusses two entirely independent embodiments, an Alpha implementation in hardware and an X86 interpreter in software.

In response, the Examiner respectfully disagrees. The cited portions by the Examiner *(col. 25, lines 11-52, col. 16, lines 1-9)* does not discuss different independent embodiments. The Examiner does not see this and the Applicant does not provide any support in the reference for this argument. The cited portions all relate to how the context data structure is used.

9. Applicant argues that Robinson deliberately avoids presenting any "extended context" that is "beyond" the capabilities of the "thread scheduler" to a "context change induced by the thread scheduler" as recited in claim 5.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., extended context that is beyond the capabilities of the thread scheduler) are not recited in the rejected

claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). It was already stated that Robinson was silent on the thread scheduler. However, Bitar teaches a thread scheduler (kernel-level scheduler 28 or User-level scheduler 26 in Fig. 3) and context switching for a multithreaded environment *(col. 12, lines 26-45, col. 13, lines 3-17, col. 7, lines 45-67 through col. 8, lines 1-8, see claim 19)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the references of Robinson and Bitar because it would increase control and will result in better scheduling decisions *(col. 5, lines 22-27)*.

10. Applicant argues that at col. 5, lines 30-34 of Bitar shows that col. 5, lines 22-27 is not motivation to combine the designated portions of Bitar and Robinson.

In response, the cited portion by the Examiner teaches that in a computer processing system, having a thread scheduler will give the benefit of increasing control for the threads and will result in better scheduling decisions. This applies motivation to have a thread scheduler applies to all computer processing systems, including the Applicant's invention. Applicant makes no arguments as the limitation not being obvious to one of ordinary skill in the art. Instead, Applicant simply argues that the motivation does not apply to the cited portions by the Examiner. However, the motivation and both Bitar and Robinson relate to a computer

processing system and there is nothing in Bitar nor Robinson that shows otherwise. Therefore, this limitation is obvious to one of ordinary skill in the art. It is noted that Applicant's citation of col. 5, lines 30-38 in Bitar actually points to another motivation to combine Robinson and Bitar.

11. Applicant states that there is no show of "reasonable expectation of success" in the Office Action.

Applicant points to MPEP 2143.02 to state the requirement of Reasonable Expectation of Success. In re Merck & Co., Inc., 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986) found that similar structure of the chemical compound was found to be considered as reasonable expectation of success (MPEP 2143.02). From the cited references by the Examiner, it is apparent that Bitar and Robinson are also similar in structure in that they share the same field of endeavor of context switching. Applicant argues that there is no reasonable expectation of success but does not provide any evidence to support his argument. A proper rebuttal requires providing evidence of superior or unexpected results (MPEP 2105, 716.02, 716.02(a) [R-2]).

12. Applicant argues that it is not shown how Bitar's scheduler with contexts be used with Robinson's contexts.

In response, Bitar and Robinson are in the same field of endeavor of context switching and Bitar does not teach away from Robinson. Evidence is not shown by the Applicant that Bitar teaches away from Robinson.

13. Applicant argues that the Examiner fails to comply with 37 CFR 1.104(c)(2). Applicant states that 37 CFR 1.104(c)(2) requires two things: (a) that "particular portions" of references

be designated, and (b) that a "clear" explanation of "pertinence" by provided. Applicant states that the Examiner consistently provide the first, but omit the second. The small parenthetical in the Office Action's discussion of claim 56 was very helpful in understanding a position.

In response, the Applicant's recitation of 37 CFR 1.104(c)(2) is not only incomplete but misleading. 37 CFR 1.104(c)(2) states that the pertinence of each reference must be clearly explained, **only if not apparent**. The Applicant's attempt to require the Examiner to give a description of every reference mapping made is incorrect under 37 CFR 1.104(c)(2) and unnecessarily burdens the Examiner. Applicant acknowledges that the Examiner has given parenthetical descriptions for various claim limitations, such as in claim 56. In the case of claim 56, for example, the reference mapping was not apparent, and therefore, the explanation was made by the Examiner.

14. Applicant argues that Robinson does not mention an exception that could correspond to the "resumption exception".

In response, the Examiner respectfully disagrees. An exception or interrupt occurs so that it is known when the resumption is to start.

15. Applicant argues that there are mixing and matching of unrelated portions of Robinson (Col. 11, lines 10-20 vs. Col. 27, lines 1-15) and that the paraphrasing is inaccurate, and therefore a fair reading of the reference as a whole is not given.

In response, the Examiner respectfully disagrees. Both Col. 11, lines 10-20 and col. 27, lines 1-15 relate to translating instructions, therefore, making them related. The Applicant's argument consists of finding any uncommon feature within the two reference citations and making a misleading statement such as they are unrelated. Applicant highlights and gives as an argument that the two cited portions are located sixteen columns away from each other. In

response to this argument, it is irrelevant how far away citations are located from each other – they are both related to translating instructions, and therefore, a fair reading of the reference as a whole is made.

16. Applicant argues that Robinson's Alpha environment (Architecture A, etc.) handles only Alpha context, and Robinson's X86 environment (Architecture B, etc.) only handles X86 and Robinson does not teach translating from non-native to native of anything other than instructions.

In response, the Examiner respectfully disagrees. The instructions refer to the architecture. There is a transfer from one architecture to a second/different architecture *(col. 2, lines 6-18, etc.)*.

17. Applicant argues that neither native dynamic link, nor native return address, nor non-native routine address do not correspond to the "linkage return address" of claim 79.

In response, the Examiner has stated that any of native dynamic link, native return address, or non-native routine address satisfies the broadest reasonable interpretation of "linkage return address". Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. It was already shown above that there is compliance with Rule 104(c)(2). During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." *In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). Applicant always has the opportunity to amend the claims during prosecution, and broad interpretation by the examiner reduces the possibility that the claim, once issued, will

be interpreted more broadly than is justified. *In re Prater*, 415 F.2d 1393, 1404-05, 162 USPQ 541, 550-51 (CCPA 1969).

18. Applicant states that Supervisory Examiner An specifically recommended an interview.

In response, this statement by Applicant's representative Mr. Boundy is very misleading. To make the record clear, Mr. Boundy had an interview with Supervisory Examiner An on July 7, 2005. The recommended interview that Mr. Boundy implies already took place on September 30, 2005. The Examiner notes that any interviews after a Final Rejection will be made according to MPEP 713.09.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

19. Claims 1-83 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. The following is indefinite:

i. In claim 1, it is unclear in the claim language whether "a pre-existing operating system" (line 2) is the same as "the operating system" (line 3) or if they are different. It is not made clear whether there are one or two operating systems in claim 1.

ii. Claim 56 is rejected for the same indefinite reasons as stated above in the rejection of claim 1.

iii. In claim 5, it is unclear in the claim language whether "a pre-existing thread scheduler" (line 2) is the same as "the thread scheduler" (line 4) or if they are different. It is not made clear whether there are one or two thread schedulers in claim 5.

iv. In claim 33, it is unclear in the claim language whether "a computer operating system" (line 2) is the same as "the operating system complementary.." (line 4) or if they are different. It is not made clear whether there are one or two operating systems in claim 33.

b. The following lacks antecedent basis:

v. Claim 1, "the operating system", line 3, etc.;

vi. Claim 5, "the thread scheduler", line 4, etc.;

vii. Claim 43, "the context", line 3;

viii. Claim 46, "the extended context", lines 2-3;

ix. Claim 47, "the extended context", line 2;

x. Claim 56, "the operating system", line 3;

xi. Claim 79, "the service", line 3;

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

20. Claims 33, 40-44, 50, 52, and 79 are rejected under 35 U.S.C. 102(e) as being anticipated by Robinson (US 6,199,095 B1).

21. As to claim 33, Robinson teaches a method (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)* comprising:

establishing an entry exception to be raised on entry to a computer operating system at a specified entry point or on a specified condition (exception handler transfers control to the entry point) *(col. 41, lines 64-67 through col. 42, lines 1-14, col. 33, lines 48-55)*;

establishing a resumption exception to be raised on resumption from the operation system when such resumption is complementary to one of the specified entries (resuming execution to entry point and a translated routine executes a return instruction to return control to its caller routine) *(col. 31, lines 9-38, col. 26, lines 49-55, col. 33, lines 29-55)*;

on detecting a specified entry to the operating system from interruption of a process executing on the computer, raising and servicing the entry exception, and then entering the

operating system to perform a service associated with the specified operating system entry (translated routine entry point and entry to interpreter) *(col. 32, lines 24-44)*; and

on detecting a complementary resumption, raising and servicing the resumption exception, and returning control to the interrupted process (resuming execution to entry point and a translated routine executes a return instruction to return control to its caller routine) *(col. 31, lines 9-38, col. 26, lines 29-55, col. 32, lines 24-44)*.

22. As to claim 40, Robinson teaches wherein the operating system is an operating system for a computer architecture other than the architecture native to the computer, unmodified for execution on the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

23. As to claim 41, Robinson teaches wherein the computer additionally executes an operating system native to the computer, and each exception is classified for handling by one of the native operating system or the operating system not native to the computer's architecture (native and non-native) *(col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

24. As to claim 42, Robinson teaches wherein operating system and the interrupted process execute in different instruction set architectures of the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

25. As to claim 43, Robinson teaches wherein the operating system and the interrupted process execute in different execution modes of the computer, and the steps to maintain the association between the interrupted process and the context are automatically invoked, without explicit software request, on a transition between the execution mode of the interrupted process and the execution mode of the operating system (background mode/system 34 and translating, etc.) *(col. 9, lines 29-67)*.

26. As to claim 44, Robinson teaches wherein the process execution mode and the operating system execution mode are two different instruction set architectures of the computer (background mode/system 34 and translating, etc.) *(col. 9, lines 29-67)*.

27. As to claim 50, Robinson teaches further comprising the step of modifying a linkage return address for the process to include information used to maintain the association *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)*.

28. As to claim 52, Robninson teaches as part of servicing the entry exception, modifying a linkage return address of the interrupted process, the return address being deliberately chosen so that an attempt to execute an instruction from the return address on return from the operating system will raise the resumption exception *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)*.

29. As to claim 79, Robinson teaches a method, comprising:

during invocation of a service routine (translated routine executes a call instruction) of a computer, passing a linkage return address (native dynamic link or "dylink" or linkage) to the service routine at which to resume execution on completion of the service (resume execution in the translated routine after the called routine has completed), the linkage return address being deliberately chosen (from the shadow stack 212) so that an attempt to execute an instruction from the linkage return address on return from the service routine will raise a program execution exception *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)*;

on return from the service routine, attempting to execute the instruction at the linkage return address ("dylnk", etc.) and raising the chosen exception *(routine return address that is pushed onto the stack by the program when it executes a call instruction, etc.) (col. 26, lines 1-13 and 49-67, col. 49, lines 1-10)*; and

after servicing the exception, returning control to a caller of the service routine (resuming execution to entry point and a translated routine executes a return instruction to return control to its caller routine) *(col. 31, lines 9-38, col. 26, lines 49-55, col. 33, lines 29-55)*.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

30. Claims 5-13, 15-18, 20-24, 26, 30-, and 82-83 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinson (US 6,199,095 B1) in view of Bitar et al. (hereinafter Bitar) (US 6,418,460 B1).

31. As to claim 5, Robinson teaches a method (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*, comprising:

managing and excuting concurrent threads of control, each thread having an associated context (non-native return stack 211 or a single register CONTEXT 105), an association between a thread and a set of computer resources of the associated context (thread associated context data structure, which also includes copies of integer registers) *(col. 25, lines 11-52, col. 16, lines 1-9)*; and

without modifying the thread scheduler to, maintaining an association between one of the threads and an extended context (shadow stack 212 and linked list of context data structures or an additional register) *(col. 25, lines 11-52, col. 16, lines 1-9)* of the thread through a context change (context switch or to change from non-native instructions into native instructions, etc.) *(col. 27, lines 19-26)*, the extended context including resources (registers, etc.) of the computer associated with the thread that are beyond those resources (additional registers, etc.) whose association with the thread.

Robinson teaches a process scheduler as a software service process which, amongst other things, is used to schedule various transactions (such as the translations, etc.) within and between the run-time and background systems *(col. 10, lines 25-56)*. However, Robinson fails to

explicitly state having a thread scheduler. Bitar teaches a thread scheduler (kernel-level scheduler 28 or User-level scheduler 26 in Fig. 3) and context switching for a multithreaded environment *(col. 12, lines 26-45, col. 13, lines 3-17, col. 7, lines 45-67 through col. 8, lines 1-8, see claim 19)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the references of Robinson and Bitar because it would increase control and will result in better scheduling decisions *(col. 5, lines 22-27)*.

32. As to claim 6, Robinson teaches wherein the thread scheduler is a component of an operating system of the computer, and further comprising:

establishing an entry exception to be raised on each entry to the operating system at a specified entry point or on a specified condition (exception handler transfers control to the entry point) *(col. 41, lines 64-67 through col. 42, lines 1-14, col. 33, lines 48-55)*;

establishing a resumption exception to be raised on a resumption from the operating system following on a specified entry (resuming execution to entry point and a translated routine executes a return instruction to return control to its caller routine) *(col. 31, lines 9-38, col. 26, lines 49-55, col. 33, lines 29-55)*;

on detecting a specified entry to the operating system from an interrupted process of the computer, raising the entry exception, and establishing the association as part of servicing the entry exception (translated routine entry point and entry to interpreter) *(col. 32, lines 24-44)*; and

raising the resumption exception, and as part of servicing the resumption exception, reestablishing the context in association with the resumed thread returning control to the

interrupted process (resuming execution to entry point and a translated routine executes a return instruction to return control to its caller routine) *(col. 31, lines 9-38, col. 26, lines 29-55, col. 32, lines 24-44))*.

33. As to claim 7, Robinson teaches wherein an exception handler for the entry exception is programmed to save a context of the interrupted process and modify the thread context before delivering the modified context to the operating system; and an exception handler for the resumption exception is programmed to restore the context saved by a corresponding execution of the entry exception handler (exception handler transfers control to the entry point) *(col. 41, lines 64-67 through col. 42, lines 1-14, col. 33, lines 48-55, see Abstract, col. 26, liens 33-67)*.

34. As to claim 8, Robinson teaches wherein the operating system is an operating system for a computer architecture other than the architecture native to the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

35. As to claim 9, Robinson teaches wherein the computer additionally executes an operating system native to the computer, and each exception is classified for handling by one of the two operating systems *(col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

36. As to claim 10, Robinson teaches wherein operating system and the interrupted thread execute in different instruction set architectures of the computer (of executing on a first

architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25, col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

37. As to claim 11, Robinson teaches wherein the operating system is in a binary code for a computer architecture non-native to the architecture of the computer (binary image conversion system converts instructions from a instruction set of a first, non native computer system to a second, different native computer system) *(see Abstract)*.

38. As to claim 12, Robinson teaches wherein the computer additionally executes an operating system native to the computer, and each exception is classified for handling by one of the two operating systems *(col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

39. As to claim 13, Robinson teaches wherein operating system and the interrupted thread execute in different instruction set architectures of the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25, col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

40. As to claim 15, Robinson teaches wherein thread scheduler and the thread execute in different execution modes of the computer, and the steps to maintain the association between the thread and the context are automatically invoked, without explicit software request, on a

transition between the thread execution mode and the thread scheduler execution mode (background mode/system 34 and translating, etc.) *(col. 9, lines 29-67, col. 2, lines 6-25, col. 41, lines 64-67 through col. 42, lines 1-14, col. 33, lines 48-55).*

41. As to claim 16, Robinson teaches wherein the thread execution mode and the thread scheduler execution mode are two different instruction set architectures of the computer (background mode/system 34 and translating, etc.) *(col. 9, lines 29-67, col. 2, lines 6-25, col. 29, lines 54-67 through col. 30, lines 1-5, etc.).*

42. As to claim 17, Robinson teaches during servicing the entry exception, saving a portion of the context of the computer, and altering the context of an interrupted thread before delivering the interrupted thread and its corresponding context to the operating system *(see Abstract, col. 26, liens 33-67).*

43. As to claim 18, Robinson teaches the step of modifying a linkage return address for resumption of the thread to include information used to maintain the association *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33).*

44. As to claim 20, Robinson (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)* in view of Bitar *(col. 12, lines 26-45, col. 13, lines 3-17, col. 7, lines 45-67 through*

col. 8, lines 1-8, see claim 19) teaches wherein the thread scheduler is an operating system for a computer architecture other than the architecture native to the computer.

45. As to claim 21, Robinson teaches wherein the computer additionally executes an operating system native to the computer, and each exception is classified for handling by one of the two operating systems *(col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

46. As to claim 22, Robinson teaches wherein operating system and the interrupted thread execute in different instruction set architectures of the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25, col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

47. As to claim 23, Robinson teaches wherein thread scheduler and the thread execute in different execution modes of the computer, and the steps to maintain the association between the thread and the context are automatically invoked, without explicit software request, on a transition between the thread execution mode and the thread scheduler execution mode (background mode/system 34 and translating, etc.) *(col. 9, lines 29-67)*.

48. As to claim 24, Robinson teaches wherein the thread execution mode and the thread scheduler execution mode are two different instruction set architectures of the computer (of executing on a first architecture or native code but accessed from a client executing in an

execution engine of a second architecture or non-native code) *(col. 2, lines 6-25, col. 9, lines 29-67).*

49. As to claim 26, Robinson teaches: in an interrupt handler of the computer, saving a portion of the context of the computer, and altering the context of an interrupted thread before delivering the interrupted thread and its corresponding context to the thread scheduler *(see Abstract, col. 26, liens 33-67).*

50. As to claim 30, Robinson teaches modifying a linkage return address for the thread to include information used to maintain the association *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33).*

51. As to claim 82, Robinson teaches without modifying a pre-existing thread scheduler of the computer, establishing an entry handler (background system 34 or the interpreter 44, or part of the exception handler, etc.) for execution at a specified entry point or on a specified entry condition to the thread scheduler (entry to second architecture from first architecture, etc.), the entry handler programmed to save a context of an interrupted thread and modify the thread context before delivering the modified context to the thread scheduler (snap-shot of the current state saved in context data structure) and modify the thread context before delivering the modified thread context to the operating system (translate from non-native to native or temporary storage for logical register manipulations, e.g.) *(col. 25, lines 10-32, col. 11, lines 10-20, col. 27, lines 1-15).*

52. As to claim 83, Robinson teaches a method, comprising:

during invocation of a service routine (translated routine executes a call instruction) of a computer, passing a linkage return address (native dynamic link or "dylink" or linkage) to the service routine at which to resume execution on completion of the service (resume execution in the translated routine after the called routine has completed), the linkage return address being deliberately chosen (from the shadow stack 212) so that an attempt to execute an instruction from the linkage return address on return from the service routine will raise a program execution exception *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)*;

on return from the service routine, attempting to execute the instruction at the linkage return address ("dylnk", etc.) and raising the chosen exception *(routine return address that is pushed onto the stack by the program when it executes a call instruction, etc.) (col. 26, lines 1-13 and 49-67, col. 49, lines 1-10)*; and

after servicing the exception, returning control to a caller of the service routine (resuming execution to entry point and a translated routine executes a return instruction to return control to its caller routine) *(col. 31, lines 9-38, col. 26, lines 49-55, col. 33, lines 29-55)*.

53. **Claims 19, 27 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinson (US 6,199,095 B1) in view of Bitar et al. (hereinafter Bitar) (US 6,418,460 B1), and further in view of Mann (US 6,154,857).**

54. As to claim 19, Robinson fails to explicitly teach wherein the modification leaves at least half of the bits of the linkage return address intact. However, Mann teaches modifying/updating at least half (one of the halves or both halves) of the data registers (col. 11, lines 10-13). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of modifying/updating at least half (one of the halves or both halves) of the data registers to the registers of Robinson because efficiency is increased by not modifying any halves that are unnecessary *(col. 11, lines 10-13)*.

55. As to claim 27, Robinson teaches wherein the operating-system-maintained resources of the thread context include data registers of the non-native computer architecture (translating between native and non-native using registers) *(see rejection of claim 5)*. Robinson fails to explicitly teach modifying at least half of the data registers. However, Mann teaches modifying/updating at least half (one of the halves or both halves) of the data registers (col. 11, lines 10-13). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of modifying/updating at least half (one of the halves or both halves) of the data registers to the registers of Robinson because efficiency is increased by not modifying any halves that are unnecessary *(col. 11, lines 10-13)*.

56. As to claim 31, Robinson fails to explicitly teach modifying at least half of the data registers. However, Mann teaches modifying/updating at least half (one of the halves or both halves) of the data registers (col. 11, lines 10-13). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of modifying/updating at

least half (one of the halves or both halves) of the data registers to the registers of Robinson because efficiency is increased by not modifying any halves that are unnecessary *(col. 11, lines 10-13)*.

57. **Claims 34-36, 38-39, 56-60, 62-67, 71-72, and 74-75 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinson (US 6,199,095 B1) in view of Fleck et al. (hereinafter Fleck) (US 6,128,641).**

58. As to claim 34, Robinson *(col. 25, lines 10-32, col. 11, lines 10-20, col. 27, lines 1-15)* in view of Fleck *(col. 1, lines 47-55, col. 2, lines 7-31)* teaches wherein an exception handler for the entry exception is programmed to save a context of the interrupted process and modify the thread context before delivering the modified context to the operating system; and an exception handler for the resumption exception is programmed. Robinson does not explicitly state to restore the context that is saved. However, Fleck teaches context switching which includes the following: context save areas *(col. 1, lines 47-55)*; saving contexts resulting from a sequence of calls, traps, or interrupts; exiting a called function or trap or interrupt handler and switching back to the previous context *(col. 2, lines 7-10 and 24-25)*; instructions to return from an interrupt or trap handler *(col. 2, lines 15-18)*; restoring the saved context (col. 2, lines 19-45); and resuming execution after exiting *(col. 2, lines 30-31)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the references of Robinson and Fleck

because it would reduce memory space and execution overhead for both task switching and for function calls and returns *(col. 1, lines 35-44)*.

59. As to claim 35, Robinson teaches wherein the operating system is an operating system for a computer architecture other than the architecture native to the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

60. As to claim 36, Robinson teaches wherein operating system and the interrupted thread execute in different instruction set architectures of the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

61. As to claim 38, Robinson teaches wherein the operating system and the process execute in two different instruction set architectures of the computer, and at least some of the steps to maintain the association between the process and the context are automatically invoked, without explicit software request, on a transition between the instruction set architectures *(col. 2, lines 6-25, col. 41, lines 64-67 through col. 42, lines 1-14, col. 33, lines 48-55)*.

62. As to claim 39, Robinson *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)* in view of Fleck *(col. 1, lines 47-55, col. 2, lines 7-31)* teaches modifying a linkage return address for the process to include information used to restore the context.

63. As to claim 56, Robinson teaches a method (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*, comprising:

without modifying a pre-existing operating system of the computer, establishing an entry handler (background system 34 or the interpreter 44, or part of the exception handler, etc.) for execution at a specified entry point or on a specified entry condition to the operating system (entry to second architecture from first architecture, etc.), the entry handler programmed to save a context of an interrupted thread (snap-shot of the current state saved in context data structure) and modify the thread context before delivering the modified context to the operating system (translate from non-native to native or temporary storage for logical register manipulations, e.g.) *(col. 25, lines 10-32, col. 11, lines 10-20, col. 27, lines 1-15)*;

without modifying the operating system, establishing an exit handler (part of the exception handler, etc.) for execution on resumption (resuming execution to entry point and a translated routine executes a return instruction to return control to its caller routine) from the operating system following an entry through the entry handler (part of the exception handler) *(col. 31, lines 9-38, col. 26, lines 49-67, col. 33, lines 29-55)*.

Robinson does not explicitly state to restore the context that is saved. However, Fleck teaches context switching which includes the following: context save areas *(col. 1, lines 47-55)*; saving contexts resulting from a sequence of calls, traps, or interrupts; exiting a called function or trap or interrupt handler and switching back to the previous context *(col. 2, lines 7-10 and 24-25)*; instructions to return from an interrupt or trap handler *(col. 2, lines 15-18)*; restoring the

saved context (col. 2, lines 19-45); and resuming execution after exiting *(col. 2, lines 30-31)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the references of Robinson and Fleck because it would reduce memory space and execution overhead for both task switching and for function calls and returns *(col. 1, lines 35-44)*.

64. As to claim 57, Robinson teaches:

scheduling concurrent threads of control by the operating system, each thread having an associated context (non-native return stack 211 or a single register CONTEXT 105), an association between a thread and a set of computer resources of the associated context being maintained by the operating system (thread associated context data structure, which also includes copies of integer registers) *(col. 25, lines 11-52, col. 16, lines 1-9)*; and

the entry and exit handlers (part of the exception handler, etc.) being programmed to maintain an association between one of the threads and an extended context of the thread through a context change induced by the operating system, the extended context including resources of the computer associated with the thread that are beyond those resources (additional registers, etc.) whose association with the thread is maintained by the operating system (context switch or to change from non-native instructions into native instructions, etc.) *(col. 27, lines 19-26)*.

65. As to claim 58, Robinson teaches wherein the operating system is an operating system for a computer architecture other than the architecture native to the computer (of executing on a first

architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

66. As to claim 59, Robinson teaches wherein the operating system and the thread execute in different execution modes of the computer, and the steps to maintain the association between the thread and the context are automatically invoked, without explicit software request, on a transition between the thread execution mode and the operating system execution mode (background mode/system 34 and translating, etc.) *(col. 9, lines 29-67)*.

67. As to claim 60, Robinson teaches in the entry handler, saving a portion of the context of the computer, and altering the context of the interrupted thread before delivering the interrupted thread and its corresponding context to the operating system *(see Abstract, col. 26, liens 33-67)*.

68. As to claim 62, Robinson teaches modifying a linkage return address for the thread to include information used to maintain the association *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)*.

69. As to claim 63, Robinson teaches wherein the operating system is an operating system for a computer architecture other than the architecture native to the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

70. As to claim 64, Robinson teaches wherein the computer additionally executes an operating system native to the computer, and each interrupt or exception is classified for handling by one of the two operating systems *(col. 29, lines 54-67 through col. 30, lines 1-5, etc.)*.

71. As to claim 65, Robinson teaches wherein operating system and the interrupted thread execute in different instruction set architectures of the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*.

72. As to claim 66, Robinson teaches wherein the operating system and the thread execute in different execution modes of the computer, and the steps to maintain the association between the thread and the context are automatically invoked, without explicit software request, on a transition between the thread execution mode and the operating system execution mode (background mode/system 34 and translating, etc.) *(col. 9, lines 29-67)*.

73. As to claim 67, Robinson teaches wherein the thread execution mode and the operating system execution mode are two different instruction set architectures of the computer (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25, col. 9, lines 29-67)*.

74. As to claim 71, Robinson *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)* in view of Fleck *(col. 1, lines 47-55, col. 2, lines 7-31)* teaches modifying a linkage return address for the thread to include information used to restore the context of the thread.

75. As to claim 72, Robinson teaches wherein the linkage return address is modified with information indicating an execution path by which, or a condition on which, execution arrived at the entry handler *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)*.

76. As to claim 74, Robinson teaches wherein the linkage return address is modified with information indicating a storage location at which at least the portion of the thread context to be modified is saved before the modifying *(col. 26, lines 49-67, col. 28, lines 13-56, col. 29, lines 18-33)*.

77. As to claim 75, Robinson teaches:

wherein the interrupted thread at the point of interruption executes in one instruction set architecture and the operating system is coded primarily in a different instruction set architecture (of executing on a first architecture or native code but accessed from a client executing in an execution engine of a second architecture or non-native code) *(col. 2, lines 6-25)*; and

further comprising the step of setting of a register to a value that specifies actions to be taken by the entry handler or exit handler to convert operands from one form to another to conform to a data storage convention of the operating system instruction set architecture (exception handler transfers control to the entry point or translate from non-native to native or

temporary storage for logical register manipulations, e.g.) *(col. 25, lines 10-32, col. 11, lines 10-20, col. 27, lines 1-15, col. 41, lines 64-67 through col. 42, lines 1-14, col. 33, lines 48-55).*

78. **Claims 45-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinson (US 6,199,095 B1) in view of Mann (US 6,154,857).**

79. As to claim 45, Robinson teaches a service routine for the entry exception and modifying data registers in association with the process by the operating system before delivering the process to the non-native operating system (see rejection of claim 33). Robinson fails to explicitly teach modifying at least half of the data registers. However, Mann teaches modifying/updating at least half (one of the halves or both halves) of the data registers (col. 11, lines 10-13). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of modifying/updating at least half (one of the halves or both halves) of the data registers to the registers of Robinson because efficiency is increased by not modifying any halves that are unnecessary *(col. 11, lines 10-13).*

80. As to claim 46, Robinson teaches wherein at least some of the modified registers are overwritten by information indicating a storage location at which at least the extended context, the extended context being the resources beyond those whose resource association with the process is maintained by the operating system, is saved before the modifying *(col. 25, lines 10-32, col. 11, lines 10-20, col. 27, lines 1-15).*

81. **Claims 61, 68-70, and 73 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinson (US 6,199,095 B1) in view of Fleck et al. (hereinafter Fleck) (US 6,128,641), and further in view of Mann (US 6,154,857).**

82. As to claims 61, 68, and 73, Robinson teaches a service routine for the entry exception and modifying data registers in association with the process by the operating system before delivering the process to the non-native operating system (see rejection of claim 33). Robinson fails to explicitly teach modifying at least half of the data registers. However, Mann teaches modifying/updating at least half (one of the halves or both halves) of the data registers (col. 11, lines 10-13). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of modifying/updating at least half (one of the halves or both halves) of the data registers to the registers of Robinson because efficiency is increased by not modifying any halves that are unnecessary *(col. 11, lines 10-13)*.

83. As to claim 69, Robinson teaches wherein at least some of the modified registers are overwritten by information indicating a storage location at which at least the portion of the thread context to be modified is saved before the modifying *(col. 25, lines 10-52, col. 11, lines 10-20, col. 27, lines 1-15)*.

84. As to claim 70, Robinson teaches wherein at least some of the modified registers are overwritten by a value that enables validation of the contents of the context *(col. 25, lines 10-52, col. 11, lines 10-20, col. 27, lines 1-15).*

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (571) 272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kt
3/27/06

SUPERVISORY PATENT EXAMINER